Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-10. (Canceled)

11. (New) Method for the secure execution of an instruction sequence of a computer application in the form of data, called typed data, comprising an identifier allowing, during the execution of said instruction sequence, an interpreter of a computer system, particularly an embedded microchip system, to identify a type of said typed data and to store the typed data and identifier in a first series of given locations in a memory of said computer system, wherein said interpreter generates, based on said identifier, additional data called type information elements, associated with each of said typed data, and stores or updates said type information elements in a second series of given storage locations corresponding one-to-one with the first series of given storage locations, in order to specify the type of these typed data, and in that during the execution of a sequence of instructions of predetermined types, said interpreter performs a continuous verification, prior to the execution of each of the predetermined instructions, of the matching between a type indicated by the instructions and an expected type indicated by said type information elements stored in said second series of storage locations, so that said execution is authorized only when there is match between said types.

2. (New) Method according to claim 1, wherein each of said type information elements comprises a string of bits stored in storage locations of said second series that correspond one-to-one with storage locations in said first series in which said associated typed data are stored, and the configuration whereof represents one of said types of typed data.

3. (New) Method according to claim 1, wherein said instructions are those of an application written in a programming language of typed data and typed object, said typed data are constituted by typed objects, in that the interpreter incorporated in said computer system is a piece of software called virtual machine that manipulates said typed object, in that said storage locations in said memory of the computer system being organized into stacks comprising a variable number of levels depending on the instruction, each level constituting one of said storage locations, said typed objects are stored in at least a first elementary stack called a data area and a second elementary stack call ed a local variable area, and in that said type information elements are distributed into two additional elementary stacks that correspond one-to-one with said first and second elementary stacks, in order to specify the type of said associated objects stored in said data and local variable areas.

4. (New) Method according to claim 1, wherein when there is no match, the execution of said instruction sequence is interrupted and replaced by the execution of instructions corresponding to pre-programmed security measures.

5. (New) Method according to claim 3, wherein said type information elements are associated with additional information elements that determine the size of said storage locations in said stacks storing said typed objects, in order to make variable the size of said stacks, based on said objects to be manipulated.

6. (New) Method according to claim 3, wherein said type information elements are associated with additional information elements called flags, in order to mark said objects that are associated with them and to indicate whether they should be saved in said stacks or can be erased.

7. (New) An embedded smart card system comprising computer data processing means and storage means for the secure execution of an instruction sequence of a computer application in the form of data, called typed data, comprising an identifier allowing, during the execution of said instruction sequence, an interpreter of said embedded system, to identify a type of said typed data and to store them in a first series

of given storage locations in a memory of said computer system, wherein said interpreter generates, based on said identifier, additional data called type information elements, associated with each of said typed data, and stores or updates said type information elements in a second series of given storage locations corresponding one-to-one with the first series of given storage locations, in order to specify the type of these typed data, and in that said interpreter comprises verification means for continuously verifying, during the execution of a sequence of instructions, prior to the execution of each of predetermined instructions of said sequence, the matching between a type indicated by the instructions and a type indicated by said type information elements, so as to authorize said execution only when there is a match between said types.

8. (New) The system according to claim 7, wherein said first series of given locations in said memory of the embedded microchip system being organized into stacks comprising a given maximum number of levels, each level constituting one of said storage locations, said typed data are stored in at least a first elementary stack called a data area and a second elementary stack called a local variable area, and in that said second series of storage locations is also organized into elementary stacks that correspond one to-one with said first and second elementary stacks.

9. (New) The system according claim 8, wherein said type information elements stored in said second series or storage locations are associated with additional information elements that determine the size of said storage locations in said stacks storing said typed data.

10. (New) The system according to claim 7, wherein said embedded system is a smart card.